nodes, which is a generic term referring to a point in a interconnected system. One type of computer network employs a client/server architecture, wherein the portions of network applications that interact with human users are typically separated from the portions of network applications that process requests and information. Often, the portions of an application that interact with users or access network resources are called client applications or client software, and the portions of an application that process requests and information are called server applications or server software. Client machines tend to run client software and server machines tend to run server software, however a server can be a client as well.

FIG. **1** illustrates a sample client/server network **10**. As one with ordinary skill in the art will readily appreciate, a client/server network is only one type of network, and a variety of other configurations, such as peer-to-peer connections, are also considered networks. In a client/server network, a plurality of nodes are interconnected such that the various nodes send and/or receive information to/from one another. As shown here, a server node **12** is interconnected to a plurality of client nodes **14** using a connection **16** such as a token ring, Ethernet, telephone modem connection, radio or microwave connection, or the like.

A computer readable medium **18**, shown here as a floppy diskette, holds information readable by a computer, such as programs, data, files, etc. As one with ordinary skill in the art will readily appreciate, computer readable medium can take a variety of forms, including magnetic storage (such as hard disk drives, floppy diskettes, etc.), optical storage (such as laser discs, compact discs, etc.), electronic storage (such as random access memory "RAM", read only memory "ROM", programmable read only memory "PROM", etc.), and the like. Certain types of computer readable medium, which are sometimes described as being nonvolatile, can retain data in the absence of power so that the information is available when power is restored.

When a group of computers are connected to one another, such as in a client/server network, a management service is typically used to organize, administer, and provide access to information and resources across the network. Management services usually access or include a collection of objects that represent a variety of things. For instance, some typical objects represent users, groups, printers, computers, and the like. In some management services, objects are organized in flat domains such as the SECURITY ACCOUNTS MANAGER ("SAM") of WINDOWS NT.

Another type of management service is organized as a synchronized hierarchal database called a distributed directory. One example of a distributed directory is the NOVELL DIRECTORY SERVICES ("NDS"), which is based on the X.500 network services protocol developed and published by the CCITT and Open Systems Interconnection Consortium. Another example of a distributed directory is ACTIVE DIRECTORY SERVICES ("ADS") by MICROSOFT. A distributed directory is an object database as opposed to the traditional relational model under Codd and Date. Usually in the context of a client/server network, a distributed directory spans and is shared by multiple networking server nodes, although a single server node can also maintain a distributed directory. While distributed directories are often used with client/server networks, they are not necessarily limited to the context of such networks. Information on the distributed directory can be created, read, modified, and shared by other nodes, such as client nodes or other server nodes, who have applicable access rights to the distributed directory.

A management service contains a collection of objects, sometimes referred to as identities, with associated attributes or properties. For example, the object **20** is a User object that represents a human user. Beyond representing users, objects represent things that humans relate to when dealing with computers. For instance, some typical objects might represent printers, print queues, files, resources, computers, and the like. In addition, objects can represent non-computer related things such as countries, companies, organizations, departments, buildings, and the like. Furthermore, objects can be organizational in nature to group other objects together. As one with ordinary skill in the art will readily appreciate, objects can represent virtually anything, whether imaginary or real.

The object **20** has a variety of associated attributes, such as "Given Name", "Last Name", "Title", etc. Each associated attribute has zero or more values. For example, the value for the property "Given Name" might be "George". An attribute is usually based on an attribute syntax. The data which can be entered as a value associated with the attribute is dictated by the attribute syntax. For instance, NDS version 4.1 includes the following syntaxes: Back Link, Boolean, Case Exact String, Case Ignore List, Case Ignore String, Class Name, Counter, Distinguished Name, E-mail Address, Facsimile Telephone Number, Hold, Integer, Interval, Net Address, Numeric String, Object ACL, Octet List, Octet String, Path, Postal Address, Printable String, Replica Pointer, Stream, Telephone Number, Time, Timestamp, Typed Name, and Unknown.

Typically, the structure of the distributed directory is governed by a schema. The schema defines the rules for adding and managing objects and attributes of objects in the distributed directory. These rules are specified through a data dictionary that provides a standard set of data types or classes from which objects can be derived or instantiated. The definitions found in the data dictionary can themselves be represented as objects. Each object in the distributed directory belongs to an object class that specifies which attributes are associated with the object. Generally, the schema is extensible so that it may be tailored to modify existing classes or add new classes.

The schema controls not only the structure of the individual objects, but also the relationship among the objects in the distributed directory. In controlling this relationship, the schema specifies subordination among object classes. That is, for every object there is a group of object classes from which subordinate objects can be formed. Objects that can contain other objects are called container objects, which are the building blocks of the distributed directory. Objects that cannot contain other objects are known as non-container or leaf objects

As shown in FIG. **2**, the objects within the distributed directory **30** are often organized in a hierarchal structure, generally in the form of a tree, where the branching points and leaves represent the objects. In this hierarchy, objects closer to the root are superior or parents to objects further from the root, which are considered subordinate or children. For instance, the object M is the parent of the child object C. Object M can also be referred to as the container to object C. The distributed directory **30** is additionally organized in partitions, as illustrated by the dashed ellipses, with each partition comprising a plurality of objects organized as a logical sub-tree. Like objects, partitions closer to the root of the distributed directory **30** are called parent partitions to those further from the root, which are called child partitions. Each partition takes the name of the root object of the sub-tree. For instance, the root object of Partition C is the object C.

Multiple replicas of the partitions are stored across the network **40**. Each replica of a partition holds the same set of